

1/10

PARTS			SUPPLIERS	
availability	name	supplier id	supplier id	quality
5	PO5	1	1	10
2	PO5	2	2	3
9	PO5	3	3	8

Table and Rank Attributes

Figure 1

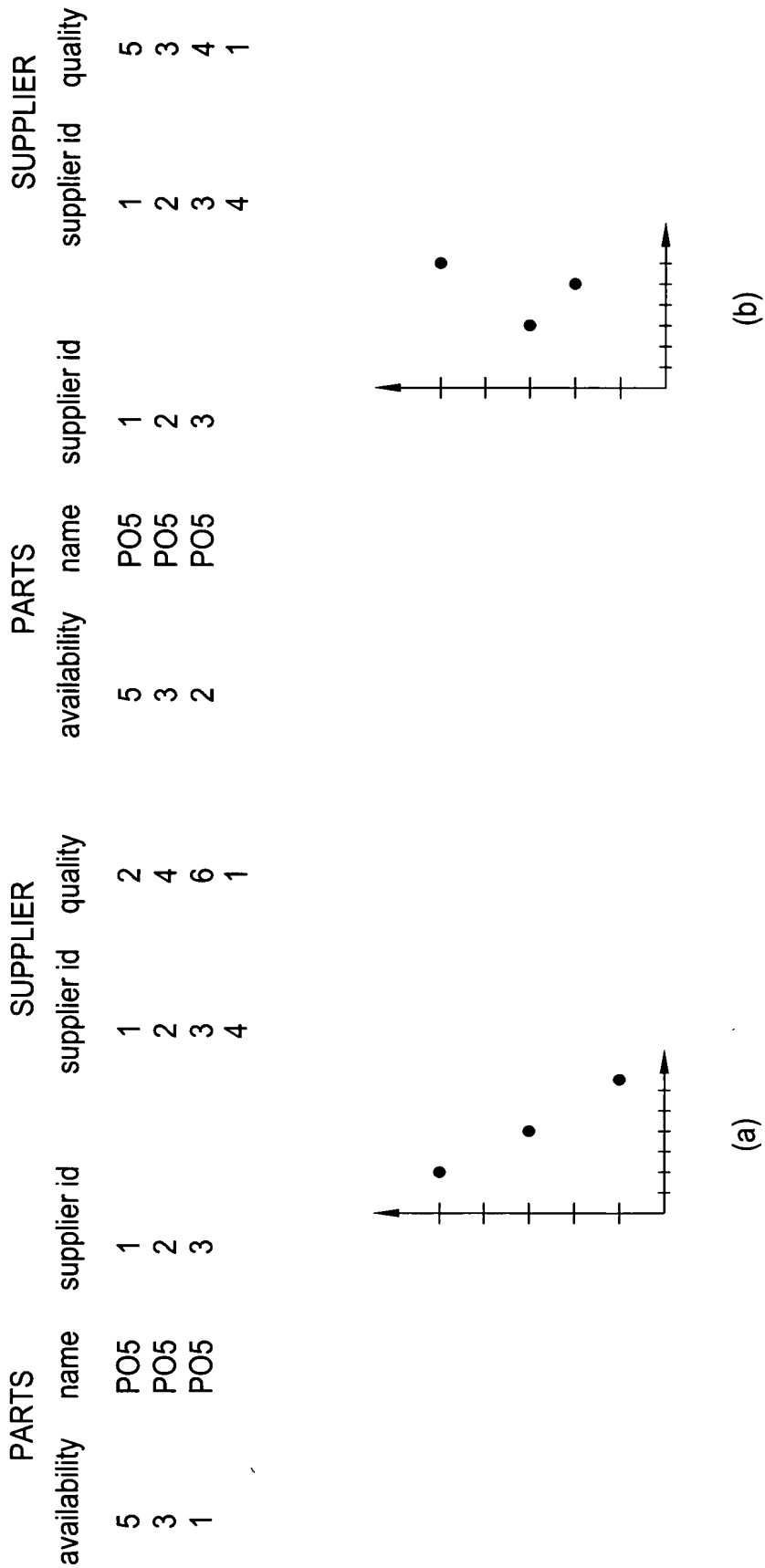
2/10

```
DominatingSet( $C, K$ )
Initialize priority queue  $Q$  and  $D_K = \emptyset$ .
Sort the join result in non-increasing order of the  $s_2$  rank value.
For the  $i$ th tuple  $t_i$  with rank values( $s_1^i, s_2^i$ )
    if ( $|Q| < K$ )
        include  $t_i$  in  $D_K$ 
        include  $s_1^i$  in  $Q$ 
    else
        if  $s_1^i \leq \min\{Q\}$  discard  $t_i$ 
    else
        include  $t_i$  in  $D_K$ 
        include  $s_1^i$  in  $Q$ 
        if  $|Q| > k$  delete the minimum element of  $Q$ 
Output  $D_K$ 
```

The Dominating Set Algorithm

Figure 2

3/10



Examples of set D_K for different join result

Figure 3

4/10

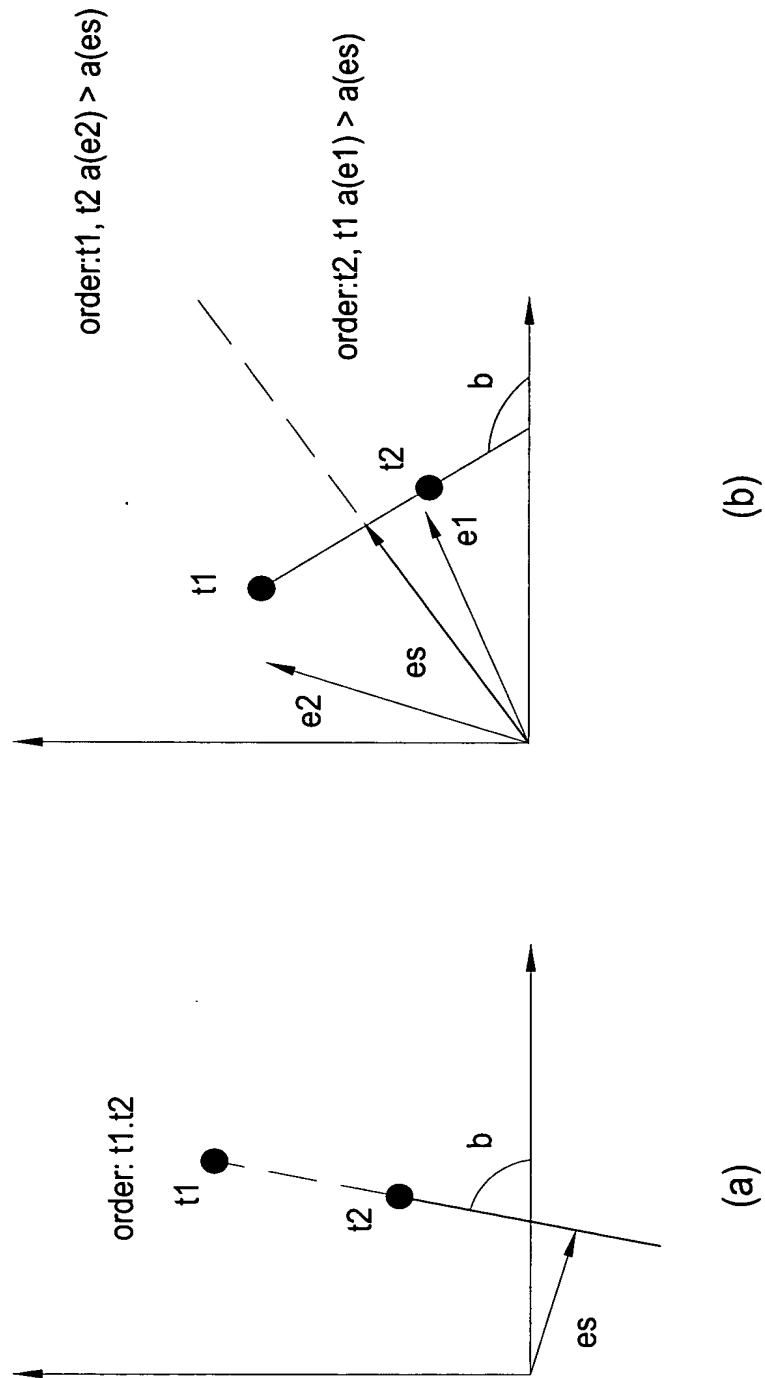


Figure 4

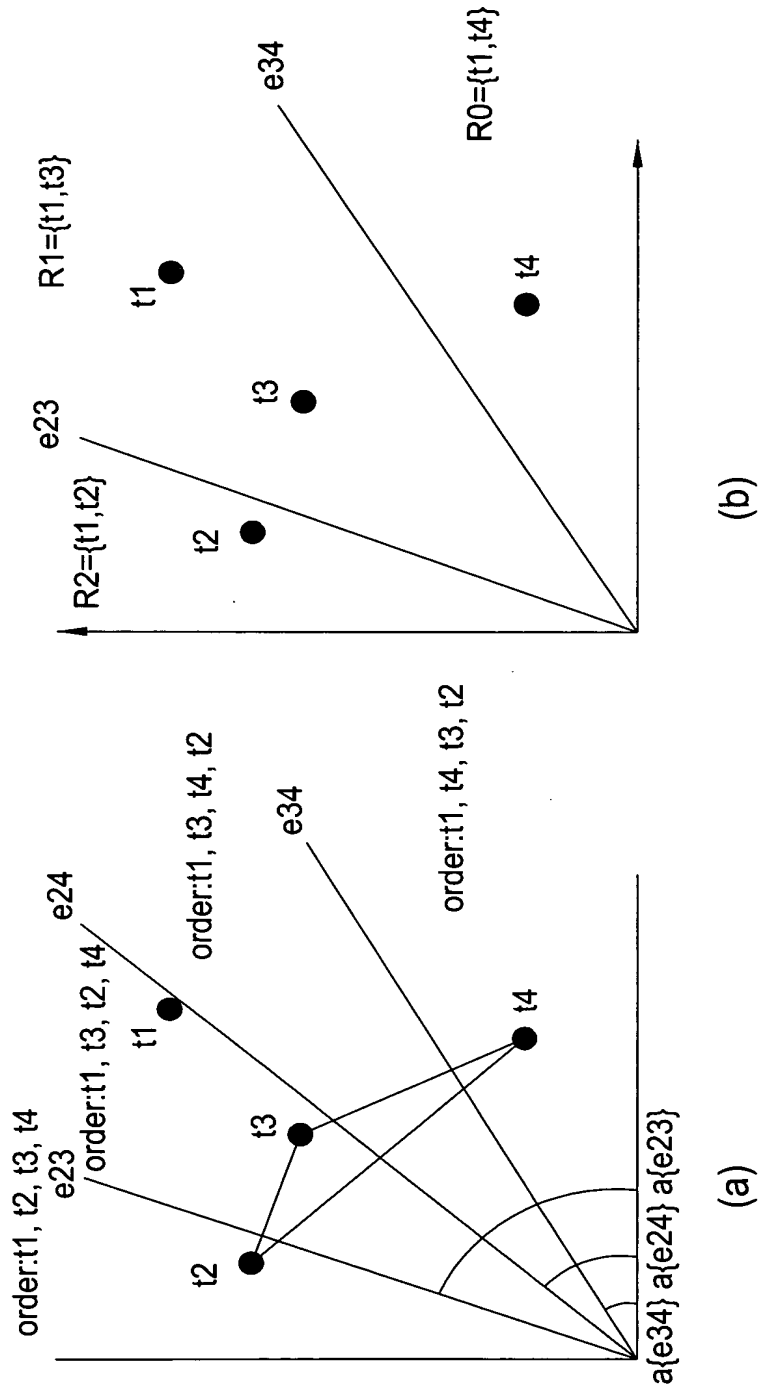
5/10

ConstructRJI(\mathcal{D}_K)
For all $(t_i, t_j), t_i, t_j \in \mathcal{D}_K$
 $V \leftarrow$ Compute separating vectors $e_{s_{ij}}$ and separating points $a(e_{s_{ij}})$
Sort V in non-decreasing order of $a(e_{s_{ij}})$
Form R consisting of top- K tuples in \mathcal{D}_K with respect to $f_{(1,0)}$
Set $\ell = 0; R_\ell = R;$
For each element (t_i, t_j) of V
 if $t_i, t_j \in R$ or $t_i, t_j \notin R$
 No change in R 's composition by $e_{s_{ij}}$; discard $e_{s_{ij}}$
 if $t_i \in R$ and $t_j \notin R$
 Materialize $a(e_{s_{ij}}), R_\ell$; replace t_i with t_j in R ;
 $\ell = \ell + 1; R_\ell = R$
 if $t_i \notin R$ and $t_j \in R$
 Materialize $a(e_{s_{ij}}), R_\ell$; replace t_j with t_i in R ;
 $\ell = \ell + 1; R_\ell = R$
When V is exhausted, materialize R

Algorithm ConstructRJI

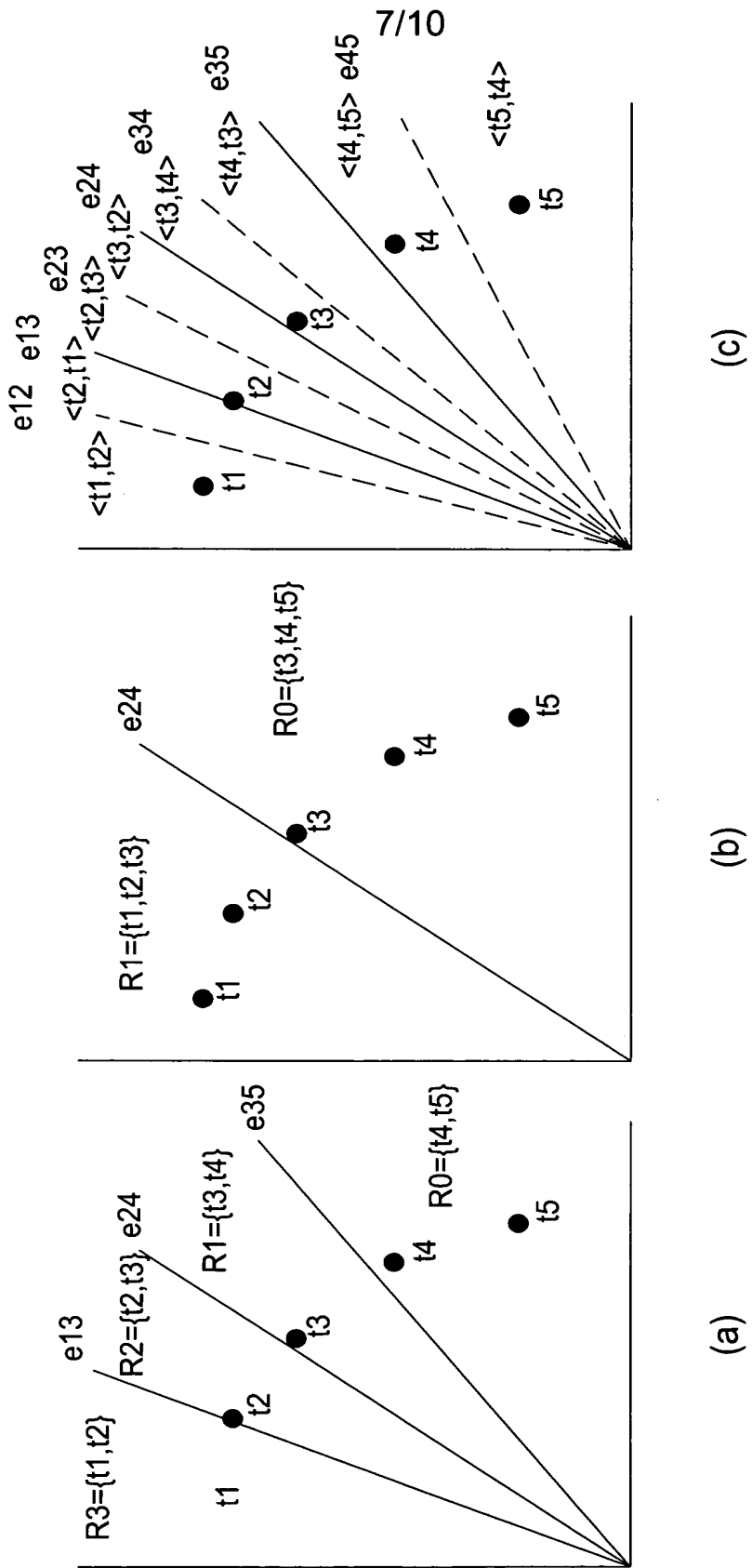
Figure 5

6/10



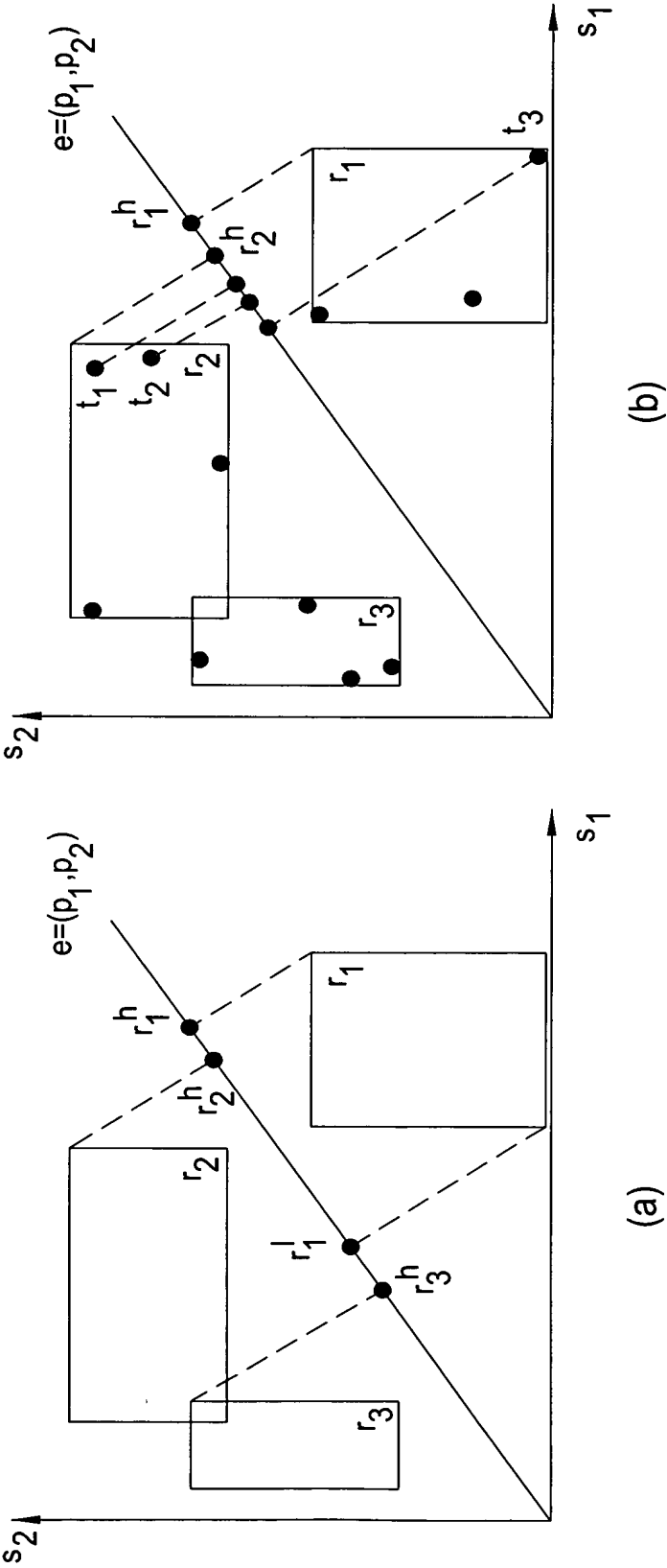
Example operation of algorithm Construct RJI

Figure 6



Space Time Tradeoffs of RJI

Figure 7



A graphical representation for the Top-KrtreeAnswer algorithm

Figure 8

9/10

Input: A number k and a preference vector $e = (p_1, p_2)$.

Output: The answer-set S to the *top-k* query.

```
1 procedure TopKrtreeAnswer()
2   let  $S = \emptyset$  be a priority queue with space for exactly  $k$  values;
3   ProcessRtreeNode(root of rtree,  $S$ );
4   return( $S$ );

5 procedure ProcessRtreeNode(node  $N$ ,  $S$ )
6   if ( $N$  is a leaf)
7     for (all tuples  $t$  in this node)
8       insert  $t$  in  $S$ ;
9   else
10    let  $r$  range over all the MBRs in  $N$ ;
11    let  $r_{max} = \arg \max_r \{ \text{maximum projection of MBR } r \text{ on preference vector } e \}$ ;
12    let  $r_{max}^{low} = \{ \text{minimum projection of MBR } r_{max} \text{ on preference vector } e \}$ ;
13    for (each subtree rooted at each MBR  $c$  of  $N$ )
14      if (maximum projection of MBR  $c \geq r_{max}^{low}$ )
15        ProcessRtreeNode( $c$ ,  $S$ );
16  return( $S$ );
```

The Top KrtreeAnswer algorithm

Figure 9

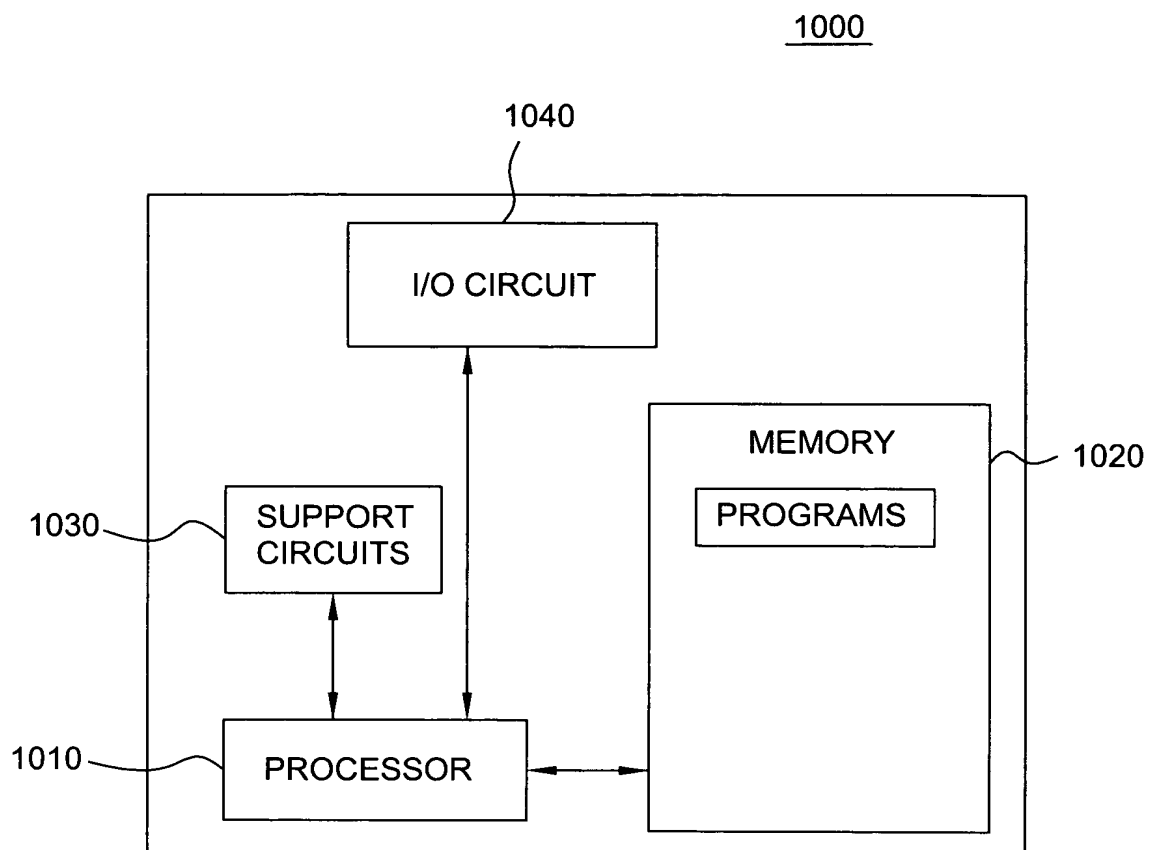


FIGURE 10